

## PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is an author's version which may differ from the publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/64350>

Please be advised that this information was generated on 2017-12-06 and may be subject to change.

# Evaluation of multimodal dialog systems

Louis Vuurpijl<sup>1</sup>, Louis ten Bosch<sup>1</sup>, Stéphane Rossignol<sup>1</sup>, Andre Neumann<sup>1</sup>,  
Norbert Pfleger<sup>2</sup>, Ralf Engel<sup>2</sup>

<sup>1</sup>NICI, The Netherlands {vuurpijl, rossignol}@nici.kun.nl, l.tenbosch@let.kun.nl

<sup>2</sup>DFKI, Germany {pfleger, rengel}@dfki.de

## ABSTRACT

This paper presents the results of an elaborate study on pen and speech-based multimodal interaction systems. The performance of the “COMIC” system is assessed through human factors analyses and evaluation of the acquired multimodal data. The latter requires tools that are able to monitor user input, system feedback, and performance of the multimodal system components. Such tools can bridge the gap between observational data and the complex process of the design and evaluation of multimodal systems. The evaluation tool presented here is validated in a human factors study on the usability of COMIC for design applications and can be used for semi-automatic transcription of multimodal data.

## Keywords

Multimodal system design and evaluation, multimodal corpora, human factors.

## Introduction

Experience has shown that the design and evaluation of multimodal interactive systems poses a complex, multi-disciplinary problem [1,2]. In large projects such as SmartKom [3] or COMIC [4], it requires a collaboration between researchers from psychology and cognitive science, up to computer science and artificial intelligence. On the one hand, the study of human subjects interacting with the system yields tons of data that can now be explored by means of “traditional” annotation and transcription tools. On the other hand, these data reveal no details about the performance of individual or mutually communicating system components on the basis of particularities in the multimodal inputs. One could state that the main problem is caused by the gap between annotating data acquired through human factors studies and using these data in the process of system design and evaluation. This paper reports on our findings in this matter in the context of the design and evaluation of the COMIC multimodal system for bathroom design.

In bathroom design, (non-expert) customers have to provide the salesperson with shape, dimensions and additional features of a bathroom. Recordings of dialogs between salespersons and customers have shown that these dialogs are inherently multimodal. In the IST project COMIC ([www.hcrc.ed.ac.uk/comic/](http://www.hcrc.ed.ac.uk/comic/)), we are developing a system that supports non-expert users with specifying the bathroom of their desire, in a way that approximates natural human-human interaction and

dialog. To build such a system, and to be able to advance our understanding of the issues involved in interaction with such a system, we need to explore how people enter data about a bathroom with pen and speech as input channels [6]. In this paper we report on an experiment that was aimed at investigating the performance of individual components of the COMIC system. To that end we performed a usability study in which naive subjects interacted with the system, and in doing so, generated a large amount of data that can be used to measure the performance of the individual system components.

Previous research (e.g. [4, 6]) has shown that it is very difficult to make sense of the data recorded in multimodal interaction systems. Even if, as is the case in the present experiment, the interaction strategy is designed to constrain the user actions, multimodal interaction appears to offer many alternative ways to approach the goal. This large degree of freedom is especially important in the analysis of interactions with naive subjects, who lack the telepathic knowledge of the system’s expectations that the system designers do have, and that helps tremendously in finding the most efficient interaction strategy and to avoid situations in which the system may not be robust. In addition, objective data (the input and output of the individual modules in a system, including time stamps attached to actions of the system and the user) form a kind of cascade. In order to analyze the performance of individual modules, for each module its complete set of input and output messages must be considered. For speech and pen input this involves manual annotation of the physical input signals. Speech input must be transcribed verbatim, as well as in the form of the concept values expressed by the words. For pen input {x,y,z} coordinate streams must be annotated with the semantic labels that are relevant in the specific application. To assess the performance of modules that have no direct relations with physical input or output, such as FUSION, which receives symbolic input of the speech and pen input processors and passes symbolic data to the dialog action manager (DAM), {input,output} pairs must also be annotated for correctness (or type of error). In the past, the development of multimodal systems has been hindered by the absence of suitable tools for annotating and analyzing interaction data. A tool for the analysis of these interaction data would greatly facilitate the evaluation of the entire dialogue system. It is the aim of this paper to present the tool that we developed to support experiments with the COMIC system.

### The COMIC System

The eventual COMIC system will comprise decoders for speech (ASR) and pen input (PII), a FUSION module that merges pen and speech input, a dialog and action manager (DAM), a Fission module that decides what information must be rendered in the form of speech, text or graphics, and output modules that generate the actual output, including an avatar with an advanced facial expression generator. The provisional system used in the experiment described in this paper had full-fledged input and fusion modules, a rudimentary DAM and simple, fixed procedures for output generation and rendering. The user interacts with the system via a head mounted close-talk microphone and a Wacom Cintiq 15X LCD tablet that acts as a paper-and-pen metaphor. COMIC employs the MULTI-PLATFORM communication architecture (MP), which is developed by DFKI, one of the partners in COMIC [5]. All data communicated between modules are encoded in XML and logged. These data provide a means for system debugging and tuning and typically are not considered when transcribing video, audio, or pen data. In the remainder of this paper, we describe our approach for combining both types of data: observational recordings and system loggings. A specification of typical multimodal system loggings and the evaluation tool “µeval” are discussed. Subsequently, we present the results from the human factors experiments that were obtained by using the new tool. We will show that µeval provides a means for semi-automatic annotation of the acquired observational data, while providing statistics on the system performance based on annotated system logs.

### General structure of multimodal system loggings

Most communication platforms like Galaxy, the Open Agent Architecture and MP provide means to log system messages. Given the multi-modular nature of multimodal systems, and because modules are typically developed by different persons, system logs can end up in a mess of messages that are only interpretable by the producer. Logs of inter-module messages nowadays are mostly encoded in XML. Messages are structured in a header, containing the source of the message, a message identifier, and timing information. The latter is extremely important and time should be synchronized over all modules. The contents of the body of a message is defined by the developers of the module that writes the message and must be parsed by all modules that read it. Loggings can become extremely large, making it very difficult to investigate failures in the communication protocols by hand. Today, no tools exist that support module developers who use MP as the integration platform in the process of debugging the distributed system messages.

The tool we developed contains knowledge about the message content and is able to parse messages produced by all current COMIC modules. It is designed such that it can monitor any message log that contains:

```
header: <timestamp> <id> <source>
body: any xml-encoded string sequence
```

For example, if a user interacting with the system would draw a wall and speak out its length, the following message sequence would be recorded:

```
<msg>t0 id0 pen-tablet
  some-sequence-of-coordinates</msg>
<msg>t1 id1 microphone
  some-audio-input</msg>
<msg>t2 id2 PII
  some-wall-encoding</msg>
<msg>t3 id3 ASR
  some-lattice-containing-length</msg>
<msg>t4 id4 FUSION
  some-wall-with-length-encoding</msg>
<msg>t5 id5 DAM
  some-rendering-and-next-state</msg>
```

In this example, it is assumed that all input data are communicated, including audio signals. In most cases however, audio and video signals do not pass through communication channels in order to reduce bandwidth. This is also the case in COMIC, where the ASR system is directly coupled to a microphone and stores audio fragments on disk. Pen coordinates are communicated and are thus contained in the multimodal system logs.

### Fast semi-automated annotation of MM interaction

When annotating multimodal interaction dialogs, the annotation process in general takes at least as long as the interaction itself. By using µeval, this process can be sped up considerably, while recording performance statistics for the individual modules. The tool considers header information present in the system logs, and sorts messages by their source and timestamp. So, messages from all PII, ASR, and other sources can easily be identified and categorized. For each message, messages from other sources that temporally correspond to it, can be detected. User input can be monitored by depicting pen input coordinates and playing audio inputs stored on disk. The latter is possible when ASR messages are marked up with the filename of the corresponding audio fragment. Now, during the processing of the recorded loggings by µeval, for each sequence of messages, the user input is rendered and the corresponding output of each module is presented in a manner that is easily readable and interpretable for a human evaluator.. The evaluator of the interaction turns can judge each output in terms of categories, such as ‘ok’, ‘false’, ‘rejected by the module’, ‘rejected by the user’, as ‘noise’, or as ‘out-of-grammar’ or ‘ignore’. All correct interpretations labeled ‘ok’ can directly be used as the label of the unknown user input, and require no further involvement of the evaluator. All other classes of input can be stored for later processing or can be transcribed manually. We have used µeval effectively for evaluating data while human factor experiments were ongoing. It appeared that the evaluation of each experiment took about 15 minutes, whereas the original interaction took on the average 60 minutes. The next sections describe the experiments and the results obtained through µeval.

### Dialog design and turn taking

Since no comprehensive taxonomy of possible speech and pen repertoires in the bathroom domain are available, it was decided to design a fully system-driven dialog. A system-driven design narrows down the set of expected user dialog acts and avoids large numbers of out-of-domain or out-of-dialog speech and pen gestures. To that end, a synchronous turn-taking protocol was developed, in which (i) the system prompts the user for information (using canned speech); (ii) the user is allowed a certain time window to enter the requested information; (iii) the input decoders process the entered information, (iv-a) the interpreted information is *beautified* or (iv-b) rejected in case the decoders cannot recognize the input.

Beautification, i.e. rendering sketches in the form of straight lines and fixed patterns, or rendering measures in ascii text, is the major way the system uses to show its interpretation of the user input. If the input can be interpreted, beautification is followed immediately by the prompt for the next information item. If the input cannot be interpreted, a more elaborate prompt is played for the previous information element.

After any system prompt, two situations can occur. If the user is satisfied with the recognition result, he can reply to the next prompt, thereby implicitly confirming the interpretation. Alternatively, the user can explicitly reject this system interpretation, either by pen or speech. One compound turn in the dialog starts with an audio prompt generated by the system, followed by a reply or reject from the user, and terminated by the interpretation (and beautification) of the system. Theoretically, all confirmed system interpretations can be used as transcription of the input [2], but in actual practice subjects accept wrong recognition results when repeated attempts to correct errors are not successful.

### Experimental design

The experiment consists of a free and a system-driven phase. In the free phase, subjects are requested to draw three bathrooms from memory, e.g., their parents', their own, and from a friend. No automatic recognition is involved. This condition serves two aims. First, natural, unconstrained, dialog acts provide essential material to further develop the various modules in the COMIC system. Second, the subjects get acquainted with the task: drawing on a tablet while using speech.

Next, they have to copy the same data into a computer system, using the tablet to sketch and write, and using speech to support their graphical input. Now, the computer does try to recognize all input gestures and utterances, using a system driven interaction strategy. Subjects are first instructed (by instructions on paper and by a video) about the automatic system. After entering the data for the three bathrooms, subjects are requested to fill in a questionnaire. In total, 28 native speaking German subjects participated with varying computer experience.

### Data collection and labeling using µeval

All logged data have been processed using our evaluation tool. For each system prompt, the expected class of user response is known (i.e. wall, window, door, or some measure). For each individual module, a label was assigned by the human evaluator to indicate the correctness of the module output ('ok','false','noise','oog'='out of grammar'). Rejects or confirmations by the user or by the system were also labeled accordingly.

All data that were interpreted by a decoder and were labeled as "ok" by the evaluator can be considered as a candidate for automatic transcription. Depending on the recognition performance of the decoding systems, this can speed up the transcription process considerably, as both segmentation and labeling are performed automatically.

Cases where the system is unable to handle the input correctly are of special interest for improvements. Also data that are rejected by the recognizer, e.g., because the user draws an unknown shape, or in cases where the user employs out-of-context speech, are interesting. For speech, these data are used to refine the language model and to tune acoustic garbage models. For pen input, these cases form examples that require new pattern recognition algorithms. Evaluators from different labs (DFKI, NICI) have used µeval for labeling and debugging purposes. It has proven to speed up both processes considerably.

### Evaluation of multimodal input

The results presented here are based on the information generated through µeval. Using the information available in the header of logged messages, the difference between two subsequent semantic expectations (broadcast by the DAM) is defined by the total turn time. Average turn time was computed for 4 input concepts and for each of the three entered bathrooms (n=28). For each concept, the average time per turn (tt), the time for recording pen inputs (tp) and speech inputs (ts) is given below. No significant decrease in turn time was observed, which indicates that subjects quickly understood the task and that the instructions they received are sufficient.

|        | Bathroom1 |     |     | Bathroom2 |     |     | Bathroom3 |     |     |
|--------|-----------|-----|-----|-----------|-----|-----|-----------|-----|-----|
|        | tt        | tp  | ts  | tt        | tp  | ts  | tt        | tp  | ts  |
| wall   | 11.4      | 4.1 | 2.9 | 11.0      | 3.6 | 3.0 | 10.5      | 3.6 | 2.8 |
| door   | 13.3      | 3.4 | 3.0 | 11.9      | 3.5 | 3.0 | 12.0      | 3.6 | 3.0 |
| window | 11.8      | 3.7 | 3.3 | 11.4      | 3.9 | 3.0 | 10.6      | 3.5 | 2.9 |
| size   | 12.2      | 3.9 | 3.4 | 11.8      | 4.1 | 3.2 | 11.8      | 3.9 | 3.2 |
| all    | 12.3      | 3.6 | 3.1 | 11.8      | 3.6 | 3.1 | 11.5      | 3.5 | 3.3 |

When considering recognition results per input category, the tables depicted below indicate whether users improve their pen and speech input over time. Since the semantic interpretation of ASR output depends on the entire recognized sentence, string error rates rather than word errors rates are reported ("zwei meter zehn" recognized as "zwei meter achtzehn" counts as one error). For sizes interpreted by PII, also string error rates (e.g., '7.13 m' incorrectly recognized as '7.18 m') are reported.

| Bathroom I |     |     |    |    | PII |     |    |    |    | ASR |   |    |    |   |
|------------|-----|-----|----|----|-----|-----|----|----|----|-----|---|----|----|---|
|            | n   | ok  | fa | r  |     | n   | ok | fa | r  |     | n | ok | fa | r |
| WALL       | 119 | 117 | 0  | 2  |     | 41  | 19 | 3  | 19 |     |   |    |    |   |
| DOOR       | 68  | 47  | 7  | 14 |     | 45  | 16 | 11 | 18 |     |   |    |    |   |
| WINDOW     | 34  | 28  | 0  | 6  |     | 40  | 22 | 7  | 11 |     |   |    |    |   |
| SIZE       | 190 | 123 | 61 | 6  |     | 201 | 66 | 81 | 54 |     |   |    |    |   |

| Bathroom II |     |     |    |    | PII |     |    |     |    | ASR |   |    |    |   |
|-------------|-----|-----|----|----|-----|-----|----|-----|----|-----|---|----|----|---|
|             | n   | ok  | fa | r  |     | n   | ok | fa  | r  |     | n | ok | fa | r |
| WALL        | 117 | 114 | 0  | 3  |     | 37  | 25 | 5   | 7  |     |   |    |    |   |
| DOOR        | 50  | 40  | 0  | 10 |     | 33  | 18 | 6   | 9  |     |   |    |    |   |
| WINDOW      | 39  | 34  | 0  | 5  |     | 34  | 22 | 2   | 10 |     |   |    |    |   |
| SIZE        | 216 | 139 | 72 | 5  |     | 219 | 84 | 105 | 30 |     |   |    |    |   |

| Bathroom III |     |     |    |    | PII |     |    |     |    | ASR |   |    |    |   |
|--------------|-----|-----|----|----|-----|-----|----|-----|----|-----|---|----|----|---|
|              | n   | ok  | fa | r  |     | n   | ok | fa  | r  |     | n | ok | fa | r |
| WALL         | 116 | 116 | 0  | 0  |     | 52  | 30 | 5   | 17 |     |   |    |    |   |
| DOOR         | 61  | 46  | 4  | 11 |     | 28  | 18 | 8   | 2  |     |   |    |    |   |
| WINDOW       | 39  | 34  | 0  | 5  |     | 28  | 20 | 2   | 6  |     |   |    |    |   |
| SIZE         | 198 | 149 | 48 | 1  |     | 235 | 89 | 109 | 37 |     |   |    |    |   |

Each row (four numbers) corresponds to respectively the total number of inputs (n), the number of correctly recognized input fragments (ok), the number of errors (fa) and the remaining (r) classes of input (rejects, noise, oog).

Recognition performance for pen input interpretation is quite well in case of the recognition of drawings. The few errors represent rather complex drawings that PII was not designed for. For sizes, it is noticeable that the performance of ASR increases in the second trial but decreases for the third bathroom. (Main factors constraining the performance of the ASR are the one-line use of the ASR, the quality of the automatic end-of-speech detection, and the used language model). Also note that there is a correspondence between the number of errors and the total number of turns. For each recognition result that is rejected by the user, the system re-phrases the question and another turn is recorded, hence the different number of inputs (n) in the tables.

### Monitoring user replies after errors

Subjects showed a variety of attitudes after an incorrect system interpretation in the speech modality. In the beginning of a test, most subjects are inclined to just repeat the utterance or repeat it slower. Rephrasing is not often used. Over sessions, the tendency to switch to the pen modality after an ASR error increases. Using the annotated system logs, such user behavior related to system responses can be monitored efficiently as below:

```
msgid expectation PII ASR FUS DAM USR
00834 WALL_LENGTH - f o drei R
00835 WALL_LENGTH - f o zwei R
00836 WALL_LENGTH - f o zwei R
00837 WALL_LENGTH o - o 3 m F
```

In this example, the user said "Drei meter" and rejected the output of ASR three times. FUSION made no errors in

passing on the interpreted inputs and only after the third try, the user switched to the pen modality, which was judged as "bk" by the evaluator, corresponding to the confirmation "F" (fixed) by the user.

### Discussion and conclusions

This paper discusses the possibility of combining the tasks of data transcription and system evaluation in one process. The approach presented here was used in a real human factors evaluation of the multimodal interaction system COMIC. Significant amounts of multimodal interaction data have been processed using the newly developed tool *µeval*. Although the tool can use many improvements, it has been validated and used effectively for system evaluation and debugging purposes. All module developers involved in input decoding (PII, ASR and FUSION) were able to browse and debug their loggings in a much more efficient way.

To our knowledge, the approach of transcribing multimodal data while annotating the corresponding session logs, has not been reported before in the literature. This approach opens up possibilities for fast transcription of observational data.

We have demonstrated that *µeval* is a flexible tool for evaluating dialogue turns in a complex human-system interaction, based on observational data and system log files. Although *µeval* is developed within the particular context of the COMIC bathroom design application and thereby implicitly makes use of the structure of the dialogue, it is basically a general-purpose tool that enables the evaluator to flexibly annotate {input, output} pairs of dialogue turns coded in XML-coded messages.

### Acknowledgements

This work is sponsored by the COMIC IST-2001-32311 project.

### References

- Oviatt, S.L., Cohen, P. R., Wu, L., et al, Designing the U-I for Multimodal Speech and Pen-based Gesture Applications: State-of-the-Art Systems and Future Research Directions in *HCI in the new millennium*, pp 419-456, 2000
- Potamianos, A., Kuo, H., Pargellis, A., et al, Design principles and tools for multimodal dialog systems, in *Proc. ESCA Workshop, IDS-99*, pp 22-24, 1999
- Wahlster, W., Reithinger, N., Blocher, A. Smartkom: Multimodal Communication with a life-like Character, *Eurospeech*, Aalborg, Denmark, 2001
- den Os, E.A and Boves, L. Towards Ambient Intel-ligence: Multimodal computers that understand our intentions, *Proc. eChallenges*, Bologna, 22 – 24, 2003.
- Herzog, G., H. Kirchmann, Poller, P. et al. MULTIPLATFORM Testbed: An Integration Platform for Multimodal Dialog Systems, *Proc. HLT-NAACL'03*, Edmonton, Canada, 2003.
- Rossignol, S., ten Bosch, L., Vuurpijl, L., et al, Human-Factors issues in multi-modal interaction in complex design tasks. *HCI International*, Greece, pp 79-80, June 2003.